

jQuery

- *JavaScript-kirjasto, joka yksinkertaistaa monia hankalia JavaScript-ohjelmoinnin asioita*
- Plussia
 - kevyt, vain yksi tiedosto -> ei vaadi useita HTTP-pyyntöjä
 - sisäänrakennettu tuki eri selaimille ja versiolle
 - hyvin dokumentoitu ja helppo oppia: <http://www.w3schools.com/jquery/> | [jQuery Learning Center](#)
 - API-rajapinnan kuvaus on kattava esimerkkeineen [jQuery API](#)
 - hyvät mahdollisuudet HTML- ja CSS-muunnoksiin (DOM-manipulation)
 - yksinkertaistaa AJAX-käsittelyjä
 - sisältää paljon valmiita animaatiota ja efektejä
 - suorituskypinen ja jatkuvasti kehittyvä
 - Runsaasti tarjolla plugineja: esim. <http://jqueryui.com/>

jQueryn lisääminen web-sivuille

- 3 tapaa: Oma kirjasto, yleinen URL ja tunnetut CDN:t
- Kaksi versiota
 - Tuotantoversio (minimaalinen ja pakattu)
 - Kehitysversio (testaamiseen ja kehittämiseen)

Oma kirjasto (Lataa kirjasto <http://jquery.com/>)

```
<body>
  <!-- jQuery -->
  <script src="js/jquery.min.js"></script>
  <!-- your own code -->
  <script src="js/oma.js"></script>
</body>
```

Yleinen URL

```
<body>
  <!-- jQuery -->
```

```
<script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<!-- your own code -->
<script src="js/oma.js"></script>
</body>
```

Tunnetut CDN:t (Content Discovery Network)

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.1.1.min.js"></scri
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js
```

"document ready"

- Ennen DOMin käsittelyä tulee varmistaa, että DOM-puu on muistissa

```
// document loaded
$(document).ready(function(){
    // jQueryn programming...
});
```

Esimerkki 0610 Tulostellaan konsoliin

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery</title>
</head>
<body>

    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
    <!-- your own code -->
    <script type="text/javascript">
        // document ready
        $(document).ready(function(){
            console.log("document ready");
        });
    </script>

</body>
</html>
```

Valitsimet (selectors)

- Valitaan elementit **ja** tehdään (action) niille jotain

```
$(selector).action()
```

Esimerkki 0602

```
<body>

  <p id="first" class="big">First paragraph</p>
  <p id="second" class="small">Second paragraph</p>

  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <!-- your own code -->
  <script type="text/javascript">
    // document ready
    $(document).ready(function(){
      // all p-element background to yellow
      $("p").css("background-color","yellow");
      // element with "first" id
      $("#first").css("font-weight","bold");
      // all elements with "small" class
      $(".small").css("font-style","italic");
    });
  </script>
</body>
```

Esimerkki 0603

```
<body>

  <p id="first" class="big">First paragraph</p>
  <p id="second" class="middle">Second paragraph</p>
  <p id="third" class="small">Third paragraph</p>
  <span>Text inside span element</span>

  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <!-- own code -->
  <script type="text/javascript">
    // document ready
    $(document).ready(function(){
      // select all element which has "big" class, id "second" and span
      $(".big, #second, span").css("background-color","yellow");
    });
  </script>
</body>
```

- Kokeile w3schoolsin [jQuery Selector](#) -testeriä
- Lista eri valitsimista [jQuery Selectors](#)
- Kaikki valitsimet jQueryn sivuilla [jQuery Selectors](#)

- Voidaan käydä läpi löydetyn (valitun) elementin lähellä olevia solmuja määritelystä kohdasta alkaen
- Ensin haetaan haluttu kohta käyttämällä valitsimia (selector)
- Tämän jälkeen liikutaan suhteessa löydettyyn kohtaan DOMissa

Tutkitaan viittauksista seuraavan HTML-koodin avulla

```
<div>
  <ul>
    <li></li>
    <li><p>Text inside paragraph</p></li>
  </ul>
</div>
```

- <div>-elementti on -elementille vanhempi (parent) ja kantaisä (ancestor) kaikelle sisältämälleen
- -elementti on vanhempi (parent) molemmille -elementeille
- molemmat -elementit ovat -elementin lapsia (child)
- ylempi -elementti on vanhempi (parent) -elementeille, -elementin lapsi (child) ja <div>-elementin jälkeläinen (descendant)
- -elementit ovat ylemmän -elementin lapsia (child), - ja <div>-elementtien jälkeläisiä (descendant)
- -elementit ovat toisilleen sisaruksia (siblings), heillä on sama vanhempi
- ...

Esimerkkejä

- Näillä esimerkeillä pääsee hyvin alkuun:
https://www.w3schools.com/jquery/jquery_traversing.asp
- Katso ainakin
 - https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_parent
 - https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_children2
 - https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_find
 - https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_find2
 - https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_siblings2

Tapahtumat

- Useimmilla DOM-tapahtumille löytyy jQuery-vastaavuudet sen omien funktioiden kautta
- Tapahtuman tapahduttua suoritetaan tapahtumaan liitetty funktio
- Tapahtuman sisällä this-objekti viittaa elementtiin, joka tapahtuman on aiheuttanut
- Tapahtumaa käsittelevä funktio saa aina tapahtumaan liitetyn objektin
- Lista tapahtumankäsittelyfunktioista esimerkkeineen:
https://www.w3schools.com/jquery/jquery_ref_events.asp
- Tapahtumat voidaan jakaa karkeasti
 - hiiritapahtumiin
 - näppäimistötapauksiin
 - lomaketapahtumiin

Hiiritapahtumia

- Ks. tarvittaessa demo mouseover- ja mouseenter-funktioiden erosta <http://api.jquery.com/mouseover/>. Mouseover- ja mouseout-funktiot koskevat myös kohteena olevan elementin sisällä olevia elementtejä.

Esimerkki 0604

Kokeile ja tutki

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery</title>
    <style>
        div {
            border:1px solid #00C;
            width:300px;
            height:300px;
            background: #d2fdee;
        }
        p {
            display: flex;
            height: 100%;
            justify-content: center;
            align-items: center;
            margin: 0;
            padding: 0;
        }
    </style>

```

```

    </style>
</head>
<body>

    <div><p>Mouse is not within this paragraph</p></div>
    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
    <!-- own code -->
    <script type="text/javascript">
        // document ready
        $(document).ready(function(){
            // mouse enter
            $("div").mouseenter(function(){
                $(this).find("p").text("mouseenter happend");
            });
            // mouse leave
            $("div").mouseleave(function(){
                $(this).find("p").text("mouseleave happend");
            });
            // mouse down
            $("div").mousedown(function(){
                $(this).find("p").text("mousedown happend");
            });
            // mouse up
            $("div").mouseup(function(){
                $(this).find("p").text("mouseup happend");
            });
        });
    </script>
</body>
</html>

```

Näppäimistötaphtumia

Esimerkki 0604

Kokeile ja tutki

```

<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery</title>
</head>
<body>
    <p></p>
    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
    <!-- own code -->
    <script type="text/javascript">
        // document ready
        $(document).ready(function(){
            // mouse enter
            $("body").keydown(function(event){
                // shows event type and which key is pressed inside p-element
                // for example: "keydown : 32", when spacebar is pressed
                $(this).find("p").text(event.type + " : " + event.which);
            });
        });
    </script>

```

```
</body>
</html>
```

Lomakkeen tapahtumia

Esimerkki 0605

Kokeile ja tutki

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
</head>
<body>
  <form id="myForm">
    <table>
      <tbody>
        <tr><td>Firstname:</td><td><input type="text" id="firstname"/></td>
        <tr><td>Lastname:</td><td><input type="text"/></td></tr>
        <tr>
          <td>Car:</td>
          <td>
            <input type="radio" name="auto" value="yes" checked>Yes
            <input type="radio" name="auto" value="no">No
          </td>
        </tr>
        <tr>
          <td>Country:</td>
          <td>
            <select id="country">
              <option value="finland">Finland</option>
              <option value="sweden">Sweden</option>
              <option value="norway">Norway</option>
            </select>
          </td>
        </tr>
        <tr><td></td><td><input type="submit" value="send"></td></tr>
      </tbody>
    </table>
  </form>

  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <!-- own code -->
  <script type="text/javascript">
    // document ready
    $(document).ready(function(){
      // highlite input-element background when it has focus
      $("input[type=text]").focus(function(){
        $(this).css("background-color", "#d6ffc9");
      });
      // set color back when not active anymore
      $("input[type=text]").blur(function(){
        $(this).css("background-color", "#fff");
      });
    });
  </script>
</body>
</html>
```

```

// listen changes, called when loses focus
$("#firstname").change(function(){
    console.log($(this).val());
});
// better use keyup -> called every time when key is back to up
$("#firstname").keyup(function(){
    console.log($(this).val());
});
// radio-button changes
$("input[type=radio]").change(function(){
    console.log($(this).val());
});
// country changes
$("#country").change(function() {
    console.log($(this).val());
});
// handle submit event
$("#myForm").submit(function(event){
    console.log("Submit happend");
    // remove default submit action -> don't send data to server
    event.preventDefault();
});
});
</script>
</body>

</html>

```

on()-funktion käyttö tapahtumien yhteydessä

- halutaan sitoa sama funktio useampiin tapahtumiin. Huomalla alla olevat mouseenter- ja mouseleave-tapahtumat.

```

$("p").on("mouseenter mouseleave",function() {
    console.log("Mouse enter or leave from paragraph");
});

```

- useammat tapahtumat halutaan liittää samaan elementtiin kätevästi

```

$("p").on({
    mouseenter: function() {
        console.log("Mouse enter");
    },
    mouseleave: function() {
        console.log("Mouse leave");
    },
    click: function() {
        console.log("Mouse clicked");
    }
});

```

- halutaan välittää tietoa tapahtumaa käsittelevälle funktiolle. Alla olevassa esimerkissä text-muuttujan kautta välitetään tietoa funktiolle.


```
// send variable to event function
$("p").on("click", {
  text: "Hello jQuery"
},function(event) {
  console.log("event data: " + event.data.text); // "event data: Hello jQuery"
});
```

- tapahtuma halutaan käynnistää itse jQuery:n trigger()-funktioilla. Alla olevassa esimerkissä tekstikappaletta voidaan klikata hiirellä, mutta sama tapahtuma saadaan laukaistua myös, kun käyttäjä klikkaa tekstikappaleen alapuolella olevaa painiketta.

```
// handle click event
$("p").on("click", function() {
  console.log("Mouse clicked.");
});
// handle button click event
$("#triggerButton").click(function(){
  // trigger above function by self
  $("p").trigger("click");
});
```

- halutaan tehdä omia kustomoitua tapahtumia.

```
// define own event
$(document).on("myCustomEvent", {
  text: "Hello jQuery"
}, function(event, arg1, arg2) {
  console.log(event.data.text); // "Hello jQuery"
  console.log(arg1); // "A"
  console.log(arg2); // "B"
});
// call own event
$(document).trigger("myCustomEvent",["A","B"]);
```

off()-funktion käyttö tapahtumankäsittelijän poistamisessa

```
$("p").off("mouseenter mouseleave");
```

Tapahtumien nimiavaruus (namespace)

Suuremmissa toteutuksissa saattaa tulla sekaannuksia, jos tapahtumia rekisteröidään "yhtenevästi". Tämä voidaan estää käyttämällä nimiavaruuksia

tapahtumankäsittelijöissä. Alla olevassa esimerkissä click-tapahtumaan on liitetty oma nimiavaruus "MyNamespace". Muista keksiä tähän oma yksilöllinen "teksti".

```
$("#p").on("click.MyNamespace", function() {  
    console.log("Mouse clicked inside paragraph.");  
});  
  
// remove click event with namespace  
$("#p").off("click.PTMNamespace")
```

DOM-manipulaatiot

- DOM-manipuloinnilla tarkoitetaan elementtien (ja attribuuttien) lisäämistä, muuttamista, korvaamista, kloonamista tai poistamista.
- https://www.w3schools.com/jquery/jquery_dom_get.asp
- Uusi elementti voidaan lisätä kohde-elementin ulkopuolelle, sisäpuolelle tai ympärille

Lisääminen kohde-elementin ulkopuolelle

- .after(), .before(), .insertAfter(), .insertBefore

Esimerkki 0606

Kokeile ja tutki

```
<!doctype html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <title>jQuery</title>  
    <style>  
        div {  
            width: 100%;  
            height: 150px;  
            border: 1px solid black;  
            background-color: antiquewhite;  
        }  
    </style>  
</head>  
<body>  
  
    <div></div>  
  
    <!-- jQuery -->  
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>  
    <!-- own code -->  
    <script type="text/javascript">
```

```

// document ready
$(document).ready(function(){
    // add p-element before div-element
    $("div").before("<p>Tekstikappale 1</p>");
    // add p-element after div-element
    $("div").after("<p>Tekstikappale 2</p>");
    // add p-element after div-element
    $("<p>Tekstikappale 3</p>").insertAfter($("<div>"));
});
</script>
</body>
</html>

```

Lisääminen kohde-elementin sisäpuolelle

- .append(), .appendTo(), .html(), .prepend(), .prependTo(), .text(), .val(), .attr()

Esimerkki 0607

Kokeile ja tutki

```

<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery</title>
    <style>
        div {
            width: 100%;
            height: 150px;
            border: 1px solid black;
            background-color: antiquewhite;
        }
    </style>
</head>
<body>

    <div></div>
    <p id="first"></p>
    <p id="second"></p>

    <ul>
        <li>First</li>
        <li>Second</li>
        <li>Third</li>
    </ul>

    <a id="link" href="http://google.com">Google</a>
    <p id="para"></p>

    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
    <!-- own code -->
    <script type="text/javascript">
        // doc ready
        $(document).ready(function(){
            // add p-element at the end of div-element
            $("div").append("<p>Paragraph 1</p>");
            // add p-element at the begining of div-element

```

```

$("div").prepend("<p>Paragraph 2</p>");
// add p-element to div-element
$("<p>Paragraph 3</p>").appendTo($("#div"));
// add content to p-element, there are two p-elements id="first", id="
$("#first").html("<i>Cursive text here...</i>");
$("#second").text("<i>Cursive text here too...</i>");
// move first li-element to last
$("ul li:first").appendTo("ul");
// show link href address
$("#para").text("Above link points to  " + $("#link").attr("href"));
});
</script>
</body>
</html>

```

Lisääminen ympärille

- .unwrap(), .wrap(), .wrapAll(), .wrapInner()

Esimerkki 0607

Kokeile ja tutki

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
  <style>
    div {
      background-color: yellow;
    }
    span {
      display: block;
    }
  </style>
</head>
<body>

  <p>Tekstikappale</p>
  <p>Toinen tekstikappale</p>
  <button id="ekaButton">Lisätään div-elementti tekstikappaleiden ympärille</but
  <br/><br/>

  <span>Tässä on tekstiä span-elementin sisällä...</span>
  <hr/>
  <span>Tässä on myös tekstiä span-elementin sisällä</span>

  <br/>
  <button id="tokaButton">Lisätään div-elementti ja kaikki span-elementit sen si
  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <!-- own code -->
  <script type="text/javascript">
    // doc ready
    $(document).ready(function(){
      // handle click event
      $("#ekaButton").click(function(){
        // wrap p-elements inside div
        $("p").wrap("<div></div>");

```

```

    });
    // handle click event
    $("#tokaButton").click(function(){
        // "move" all span-elements inside a new div
        $("span").wrapAll("<div></div>");
    });
});
</script>
</body>
</html>

```

Elementtien korvaaminen ja kloonaminen

- .replaceAll(), replaceWith(), .clone()

Esimerkki 0608

Kokeile ja tutki

```

<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery</title>
</head>
<body>

    <p>Sample text here</p>
    <p>Sample text here</p>
    <p>Sample text here</p>

    <button id="firstButton">Replace all p-elements to h1-elements</button>
    <br/>
    <br/>
    <button id="secondButton">Replace back to p-elements</button>

    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
    <!-- own code -->
    <script type="text/javascript">
        // doc ready
        $(document).ready(function(){
            // replace elements
            $("#firstButton").click(function(){
                $("<h1>p-elements replaced to h1-elemens</h1>").replaceAll("p");
            });
            // replace elements
            $("#secondButton").click(function(){
                $("h1").replaceWith $("<p>Sample text here</p>");
            });
        });
    </script>
</body>
</html>

```

Kokeile ja testaa (Huomaa lisätä itse PICTURE.png)

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
</head>
<body>

  <br/>
  <button id="cloneButton">Clone Me!</button>
  <hr/>
  <div id="clones"/>
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function(){
      // take a clone only from the first one
      $("#cloneButton").click(function(){
        $(".clone:first").clone().appendTo($("#clones"))
      });
    });
  </script>
</body>
</html>

```

Elementtien poistaminen

- .detach(), .empty(), .remove(), .unwrap()

Esimerkki 0609

Kokeile ja tutki

```

<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
</head>
<body>
  <h1>Random numbers</h1>
  <ul>
    <li>0.11428727184306786</li>
  </ul>
  <button id="addButton">Add a new number</button>
  <button id="removeButton">Remove all</button>
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function(){
      // add a new number
      $("#addButton").click(function(){
        $("ul").append("<li>"+Math.random()+"</li>");
      });
      // remove all numbers, leave ul-list
      $("#removeButton").click(function(){
        $("ul").empty();
      });
    });
  </script>
</body>
</html>

```

Elementtien CSS-ominaisuuksien muuttaminen

- .css(), .addClass(), .hasClass(), .removeClass(), .toggleClass()

Esimerkki 0610

Kokeile ja tutki

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
  <style>
    .important {
      font-weight: bold;
      color: red;
    }
  </style>
</head>
<body>
  <h1>CSS changes</h1>
  <p>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </p>
  <p id="target">
    Lorem Ipsum is simply dummy text of the printing and typesetting industry.
  </p>
  <button id="first">Add CSS</button>
  <button id="second">Remove CSS</button>
  <button id="third">Toggle CSS</button>
  <button id="fourth">Add CSS by code</button>
  <button id="fifth">Remove CSS by code</button>
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function(){
      // add css class
      $("#first").click(function(){
        $("#target").addClass("important");
      });
      // remove css class
      $("#second").click(function(){
        $("#target").removeClass("important");
      });
      // toggle on/off css class
      $("#third").click(function(){
        $("#target").toggleClass("important");
      });
      // add css by coding
      $("#fourth").click(function(){
        $("#target").css("font-style","italic");
      });
      // remove coding made cscs
      $("#fifth").click(function(){
        $("#target").css("font-style","");
      });
    });
  </script>
</body>
</html>
```

Efektit

- elementtejä voidaan piilottaa `.hide(aika, funktio)` tai tuoda esille `.show(aika, funktio)` hidastetusti ja toiminnon jälkeen voidaan kutsua haluttua funktiota. Funktiolla `.toggle(aika, funktio)` elementti piilotetaan tai tuodaan esiin riippuen siitä, onko elementti esillä.
- aika = "slow", "fast" tai millisekuntti
- `.hide()`-funktio asettaa elementin CSS display -ominaisuuden arvoon none ja tällöin elementin leveys ja korkeus asettuu arvoon nolla

Esimerkki 0611

Kokeile ja tutki

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
  <style>
    div {
      border:1px solid #00C;
      width:300px;
      height:150px;
      background: #d2fdee;
    }
    p {
      display: flex;
      height: 100%;
      justify-content: center;
      align-items: center;
      margin: 0;
      padding: 0;
    }
  </style>
</head>
<body>

  <div id="myDIV">
    <p>Click below buttons!</p>
  </div>
  <br/><br/>
  <button id="showButton">show()</button>
  <button id="hideButton">hide()</button>
  <button id="toggleButton">toggle()</button>
  <br/><br/>
  <button id="showButton2">show(2000,ready)</button>
  <button id="hideButton2">hide(2000,ready)</button>
  <button id="toggleButton2">toggle(2000,ready)</button>

  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function(){
      // show
      $("#showButton").click(function(){
        $("#myDIV").show();
```



```

    });
    // hide
    $("#hideButton").click(function(){
        $("#myDIV").hide();
    });
    // toggle
    $("#toggleButton").click(function(){
        $("#myDIV").toggle();
    });

    // show 2
    $("#showButton2").click(function(){
        $("#myDIV").show(2000,ready);
    });
    // hide 2
    $("#hideButton2").click(function(){
        $("#myDIV").hide(2000,ready);
    });
    // toggle 2
    $("#toggleButton2").click(function(){
        $("#myDIV").toggle(2000,ready);
    });

    });

    function ready() {
        console.log("Animation is ready!");
    }
</script>
</body>
</html>

```

Haihdutus ja liu'utus (fade, slide)

- `slideDown(aika,funktio)`, `slideUp(aika,funktio)`, `slideToggle(aika,funktio)`,
`fadeIn(aika,funktio)`, `fadeOut(aika,funktio)`,
`fadeTo(arvo,aika,funktio)`, `fadeToggle(aika,funktio)`

Esimerkki 0612

Kokeile ja tutki

```

<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>jQuery</title>
    <style>
        img {
            opacity: 0.5;
        }
    </style>

</head>
<body>

    <div>
        
        
        
    </div>

```

```

</div>

<script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<script>
// document ready
$(document).ready(function(){
    // show image 100%
    $("img").mouseover(function(){
        $(this).fadeTo("fast",1.0);
    });
    // fade image 50%
    $("img").mouseout(function(){
        $(this).fadeTo("fast",0.5);
    });
});
</script>

</body>
</html>

```

Toiminnassa toistaiseksi täällä:

<http://student.labranet.jamk.fi/~ara/examples/fade/>

Animaatiot

- `.animate({objekti}, aika, funktio)` -funktio

Tutustu seuraaviin esimerkkeihin

- https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation1_multicss
- https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation1_relative
- https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation1_toggle
- https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation
- https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_animation2

jQuery ja AJAX

Aiemmin esitettiin tiedon lataamisen perusteita JavaScript-ohjelmointia käyttämällä. DOM-manipulaatiot ja AJAX-käsittelyt ovat kuitenkin huomattavasti suoraviivaisempia jQueryn avulla. Ota huomioon jQueryn 3-versio on muuttanut AJAXiin liittyviä toimintoja merkittävästi.

Esimerkki 0612 - ladataan ja esitetään JSON-tietoa

1. Käytetään friends.json-tietoa

```
{
  "friends":
  [
    { "id":"id1", "firstName":"Jyrki", "lastName":"Kernel", "email":"jyrki@kernel.
    { "id":"id2", "firstName":"Anna", "lastName":"Husso", "email":"anna@husso.com"
    { "id":"id3", "firstName":"Josef" , "lastName":"Staffer", "email":"josef@staff
  ]
}
```

2. Luodaan HTML-tiedosto, jossa käytetään jQueryä

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>jQuery - AJAX</title>
  </head>
  <body>
    <ul id="friendsList"></ul>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min
    <script src="load.js"></script>
  </body>
</html>
```

3. Luodaan load.js JavaScript-tiedosto, joka käyttää jQueryä ajax-käskyjen hallinnassa

```
$(document).ready(function(){
  $.ajax({
    url: "friends.json",
    cache: false
  }).done(function(data) {
    console.log("done");
    showFriends(data);
  }).fail(function() {
    console.log("error");
  }).always(function() {
    console.log("complete");
  });
});

function showFriends(data) {
  $.each(data.friends, function(index, friend) {
    $("#friendsList").append("<li>" + friend.firstName + " "
                                + friend.lastName + " "
                                + friend.email
```

```
    + "</li>");  
  });  
}
```

jQuery.ajax()-metodin toiminnallisuuksia

- <https://learn.jquery.com/ajax/jquery-ajax-methods/>

Useimmin käytettyjä ovat

- **url:** osoite, jonne AJAX-pyyntö lähetetään
- **async:** aseta arvoon false, jos haluat käyttää synkronista latausta (oletus true)
- **beforeSend:** mahdollisuus toteuttaa ko. funktion sisältö ennen pyyntöä
- **cache:** käytetäänkö välimuistia (oletus) vai ei
- **data:** palvelimelle lähetettävä tieto (objekti tai query-string 'etunimi=pekka&sukunimi=peloton')
- **dataType:** palvelimelta odotetun tiedon muoto: 'xml', 'html', 'script', 'json', 'jsonp', 'text'
- **method:** palvelun yhteydessä käytetty toiminto: 'GET', 'POST', 'PUT'
- **timeout:** pyyntöön liitettävä aikaraja millisekunteina

Ajax-toiminnallisuuteen liitettyjä takaisinkutsufunktioita ovat

- `.done(data, textStatus, jqXHR)` AJAX-käskey käsitelty onnistuneesti palvelimella, palautettu tieto on data-muuttujassa
- `.fail(jqXHR, textStatus, errorThrown)` AJAX-käsittelyn suhteen on esiintynyt virhettä palvelimella
- `.always(data|jqXHR, textStatus, jqXHR|errorThrown)` tämä funktio suoritetaan aina

Linkkejä

- [jQuery AJAX Methods @ W3CShools](#)
- [jQuery.ajax\(\)](#)

Jätetty tarkoituksella tyhjäksi.